Blockchain-Based Educational Certificate Verification using AES and Elliptic-curve Diffie–Hellman (ECDH)

Group: Pixel Protectors

Anudeep Gadi

Kylee Willis

Adithya Chittajallu

Naveen Kumar Attaluri

Jyothi Krishna Mannava

Rahul Karthik Arunachalam Usharani

School of Computing and Engineering: University of Missouri – Kansas City

CS 5533 Applied Cryptography

Sravya Chirandas

Date: 12/11/2023

Table of Contents

Abstract 3

Introduction 4

Background and Related Technologies 5

- Blockchain 5
- Advanced Encryption Standard (AES) 6
- Elliptic-curve Diffie–Hellman 7

Implementation 9

Advantages 16

Challenges 17

Code 18

Outputs 21

Results 23

Conclusion 24

References 25

Abstract

Ensuring the authenticity of academic credentials is crucial for higher education and immigration. The education and credentialing industry faces challenges like 'credential fraud' (fake diplomas and manipulated transcripts) and 'verification inefficiency’ (Inconsistent Record-Keeping, and difficulties in global credential validation). Eligible candidates may lose opportunities when job positions or admissions are awarded to individuals with fraudulent credentials, diminishing the chances of genuinely qualified candidates. Unintentional recruitment of ineligible candidates can impact institutions' and companies' work quality and potentially tarnish their reputation. This paper aims to address these issues by utilizing Ethereum-based blockchain technology, enhanced by the execution of smart contracts, to store academic records, Elliptic-curve Diffie–Hellman (ECDH), and AES to ensure the security and integrity of the data. These combined technologies ensure the utmost security and integrity of academic records in the digital era.

Keywords: Ethereum-based blockchain, Elliptic Curve Cryptography (ECC), Diffie-Hellman, AES

Introduction

In the context of verifying academic credentials, ensuring the legitimacy and accuracy of educational qualifications carries profound importance. This validation process is important for educational institutions to guarantee the honesty and reliability of academic certificates. However, the traditional methods used to verify these credentials have proven insufficient, giving rise to several complex challenges, notably credential fraud and inefficient verification processes. In response to these challenges, the integration of cutting-edge technologies has emerged as a promising solution. This paper examines the collaborative use of blockchain technology, Advanced Encryption Standard (AES) encryption, and Elliptic Curve Cryptography (ECC) with the Diffie-Hellman key exchange protocol to fundamentally transform the landscape of academic credential verification. This innovative approach aims to enhance security, efficiency, and transparency in the verification process, effectively addressing the issue of credential fraud.

This solution leverages the strengths of three distinct technologies. Blockchain (Ethereum) operates as an immutable digital ledger distributed across a decentralized network of interconnected computers. Each academic record is stored within an individual "block," forming an unchangeable chain that cannot be tampered with or manipulated. This not only guarantees the security of academic records but also facilitates transparency in the verification process. Advanced Encryption Standard (AES) encryption is employed to encrypt the data in the blockchain blocks. AES transforms these records into impenetrable digital vaults, ensuring that only individuals possessing the unique AES secret key can access and decipher the information. The AES encryption layer adds extra protection, by preventing unauthorized access to sensitive academic records. Elliptic-curve Diffie–Hellman (ECDH) ensures secure key exchange and communication between the educational institution (verifier) and the student. This cryptographic method establishes a secure channel for the exchange of the AES secret key. Together, these technologies create a system that is not only highly secure but also efficient, offering a streamlined and transparent approach to storing and accessing academic records.

Background and Related Technologies

The design of the system revolves around the synergy of blockchain, Advanced Encryption Standard (AES), and Elliptic-curve Diffie–Hellman (ECDH). Each technology is thoughtfully integrated for specific advantages: AES will be used to encrypt the data being stored with blockchain, and ECDH will be used to transmit the keys to authorized users. This will ensure that every step of the transfer and verification process is confidential and securely managed.

Ethereum – an opensource Blockchain

Blockchain, with a specific emphasis on Ethereum, stands as a foundation for the system due to its unique attributes. Ethereum's architecture is supported by a decentralized, peer-to-peer network of nodes that collectively validate transactions and maintain the blockchain ledger. Its architecture is composed of several key components, including the Ethereum Virtual Machine (EVM), which executes smart contracts and decentralized applications (DApps). The network's native cryptocurrency, Ether (ETH), is used to incentivize miners who secure the network and validate transactions. Ethereum employs a consensus mechanism called Proof of Stake (PoS) to secure the network, moving away from the energy-intensive Proof of Work (PoW) that blockchain networks have historically used. Proof of Stake (PoS) is a consensus mechanism in blockchain networks where validators create and add new blocks based on the amount of cryptocurrency they hold and are willing to "stake" as collateral, making it a more energy-efficient alternative to Proof of Work (PoW). In contrast, Proof of Work (PoW) relies on miners who solve complex computational puzzles to validate and add new blocks, a process known for its high energy consumption. This architectural shift aims to enhance scalability, reduce environmental impact, and ensure the network's sustainability for the long term.

Ethereum provides an immutable, decentralized ledger, ensuring secure and transparent academic record storage. Its resistance to tampering is pivotal in preserving the integrity and authenticity of academic records. Moreover, Ethereum's distributed nature enhances transparency and trust, crucial for academic credential verification. Ethereum, in particular, is favored for its capability to host smart contracts, further enhancing the automation and trustworthiness of transactions within the academic record verification process. This feature contributes significantly to the efficiency and reliability of the system, streamlining the verification of academic credentials while ensuring data security and integrity (P. Gugnani, W. W. Godfrey & D. Sadhya, 2022).

Advanced Encryption Standard with Galois/Counter Mode (GCM)

AES-256 (Advanced Encryption Standard with a 256-bit key length) is a symmetric key cipher. It is known for its exceptional cryptographic strength and versatility. This choice of using AES-256 for this proposed system is supported by the profound advantages it offers, notably its capability to ensure the utmost confidentiality and integrity of academic records. With AES-256, a critical layer of protection is introduced, improving the system's resilience against unauthorized access and data breaches. Galois/Counter Mode (GCM) employs a nonce for uniqueness, a cryptographic salt for added randomness, and a tag for data integrity verification, collectively enhancing the security of encrypted information in the system. The key advantage of AES-256 lies in its significantly increased key length, which provides an exponential increase in possible encryption combinations compared to AES-128. This enhances data security by making it extremely difficult for attackers to break the encryption through brute-force methods due to the complexity of AES-256 (D. R. Rv, S. Mohamed Thawfiq, U. Arshad Uvais, & S. Revathy, 2022).

However, there is a tradeoff in this selection. While AES-256 offers superior security, it may incur a nominal tradeoff in speed compared to AES-128 due to its more complex encryption process. The tradeoff is nonetheless justified by the heightened security it provides, which is especially important for safeguarding sensitive academic information in a transparent and immutable technology like blockchain. Additionally, AES is considered quantum-safe due to its key size and the time required for a quantum attack to compromise its security. When it comes to AES-128, the key size is relatively smaller, making it less secure, as a quantum attack could potentially break it within 2^64 operations, which is not very secure in the context of quantum computing capabilities. In contrast, AES-256 employs a larger key size, significantly improving its quantum resistance. A quantum attack on AES-256 would require a substantially longer timeframe, approximately 2^128 operations, which is still considered very secure. Therefore, AES-256 is preferred because of its enhanced key size and resistance making it a more quantum-safe choice.

Elliptic-curve Diffie–Hellman

ECDH (Elliptic-curve Diffie–Hellman) is a fundamental component in our system, providing a robust mechanism for secure key exchange between the student and the receiving institution. This key exchange process is paramount in safeguarding the confidentiality and privacy of sensitive academic record details, such as the academic record block number within the Ethereum network and the secret key necessary for decryption. The strength of ECDH is rooted in its use of elliptic curve cryptography, which employs the mathematical complexity of solving the elliptic curve discrete logarithm problem to create secure and efficient key exchanges. ECDH's security is amplified by its utilization of public and private keys. The public key is shared openly, allowing the student and the receiving institution to derive a shared secret key independently. This shared secret key is pivotal in securing data communication during the transmission process, ensuring that academic records remain confidential. ECDH is known for its computational efficiency, requiring fewer resources than some other public-key cryptographic algorithms. Each educational institution may need to verify thousands of academic records, a task that demands significant computational power. ECDH's efficiency enables the system to handle these verification processes with a lighter resource footprint, ensuring that the verification procedures are swift and resource-efficient. This is particularly valuable in resource-constrained environments. Also, ECDH is highly scalable, accommodating various key lengths to suit different security levels, offering flexibility in tailoring the level of security to specific needs. Academic institutions frequently deal with a wide range of academic records, each potentially requiring a different level of security. The ability to accommodate various key lengths allows for the flexible adaptation of security levels to the specific requirements of each academic record (T. Adhikari, M. Kule & A. K. Khan, 2022).

The amalgamation of these technologies forms the bedrock of the system, systematically addressing the challenges of academic credential verification. Blockchain guarantees secure record storage, AES bolsters data confidentiality, and ECDH facilitates secure communication. Together, they culminate in an integrated approach that guarantees the integrity, confidentiality, and privacy of academic records at every stage of the verification process. This ensures a comprehensive solution that aligns with the evolving needs of the academic ecosystem and the imperatives of data security.

Implementation

Step 1: Creating a Solidity Smart Contract (Solidity File)

Before proceeding with academic record encryption and integration with the Ethereum blockchain, it is essential to establish a secure and standardized method for recording and retrieving student data on the blockchain. This is achieved by creating a Solidity smart contract, which acts as the backbone for storing academic records securely.

Figure 1

Pragma Solidity Code

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.6 <0.9.0;
contract StudentRecord {
    struct StudentData {
        string encryptedText;
    }
    StudentData public studentRecord;
    function addRecord(string memory _encryptedText) public {
        studentRecord = StudentData({
            encryptedText: _encryptedText,
        });
    }
    function getRecord() public view returns (string memory, bytes32) {
        return (studentRecord.encryptedText);
    }
}
```

The provided Solidity contract, named "StudentRecord," serves as the cornerstone of this process (see Figure 1). It defines a structured data format for storing encrypted academic records. These records are accessible by authorized parties, ensuring data privacy and security. The contract is comprised of two core functions:

addRecord: This function allows authorized entities, such as universities, to add academic records securely to the Ethereum blockchain. It accepts encrypted text which adds a new record to the blockchain, creating an immutable and transparent ledger.

getRecord: Authorized parties, including the student or receiving institutions, can access the academic record securely by invoking this function. It returns the encrypted text, thereby enabling the retrieval of academic records.

By integrating this Solidity contract into the system, the entire process attains a solid foundation for the secure and immutable storage of academic records on the Ethereum blockchain. This addition to the implementation sequence ensures the longevity, transparency, and integrity of academic records, setting the stage for the subsequent encryption, key exchange, and secure data transmission steps.

Step 2: Record Encryption

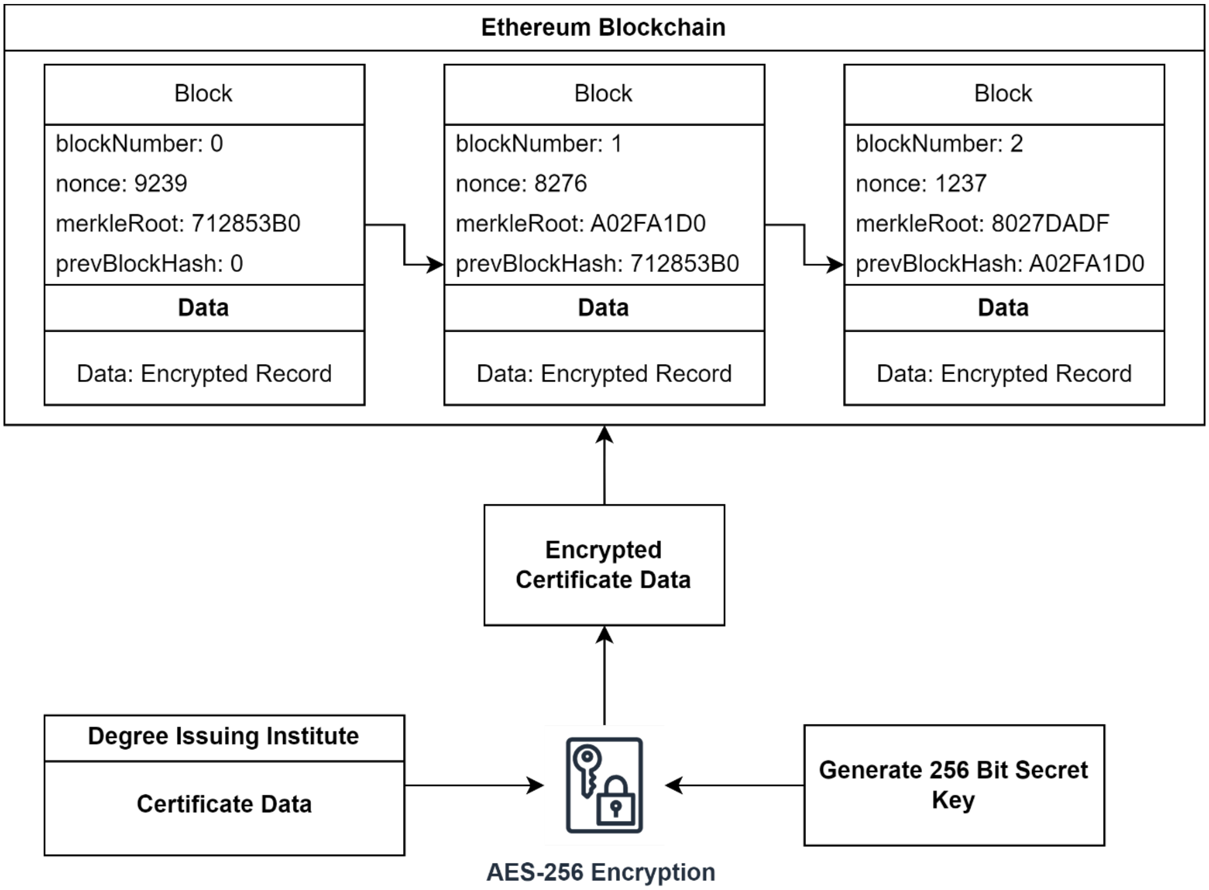
The process starts with the university generating a digital academic record for the student which can be structured in formats like JSON or CSV format. For each academic record, a unique secret key is created, and this secret key is used to encrypt the record using the Advanced Encryption Standard (AES-256 GCM) algorithm. The encrypted record is now prepared for storage on the Ethereum blockchain as shown in Figure 2. The salt, nonce, and tag generated during encryption must be passed to the student as it is required in further steps for decryption.

Step 3: Blockchain Integration

Following the encryption of academic records, the next crucial phase in the process is their secure integration with the Ethereum blockchain, as shown in Figure 2. This step is fundamental to establishing an immutable and transparent repository for academic records, ensuring their long-term security and accessibility.

Figure 2

Adding Academic Records to Ethereum Blockchain



To facilitate this integration, the system relies on two essential components: the contract's ABI (Application Binary Interface) and the contract address. The ABI serves as the interface definition that informs the system how to interact with the Ethereum smart contract, specifying the functions it exposes and the data types it expects. The contract address, on the other hand, is the unique identifier of the smart contract on the Ethereum network, pointing to the location where academic records will be stored. We use the contract's ABI and address to establish a connection with the Ethereum smart contract. This connection enables the system to access the "addRecord" function, allowing academic records to be securely appended to the blockchain. This integration sets the foundation for the subsequent phases of the process, where the academic records can be retrieved and verified by authorized entities. After the addition of the record, the associated block number and should be shared with the student for future reference using a secured communication medium.

Step 4: Key Exchange with ECC and Diffie-Hellman

The process begins with both the student, who is the data sender, and the receiving institution, the data recipient, generating their unique public and private keys. These keys serve as the foundation for the ECDH protocol. Both parties mutually decide on an elliptical curve and the public keys are openly shared between the two parties, allowing them to perform the cryptographic calculations necessary for secret key derivation. The beauty of ECDH lies in its ability to generate a shared secret key without ever revealing it in the process, thus ensuring the utmost confidentiality. The resulting shared secret key becomes a powerful encryption key.

The foundational concept behind ECDH is grounded in the following property of EC points:

(studentPrivKey * G) * institutionPrivKey = (institutionPrivKey * G) * studentPrivKey

In the proposed academic credential verification system, ECDH unfolds as follows:

G is the mutually agreed point on the elliptical curve that is set as the base point.

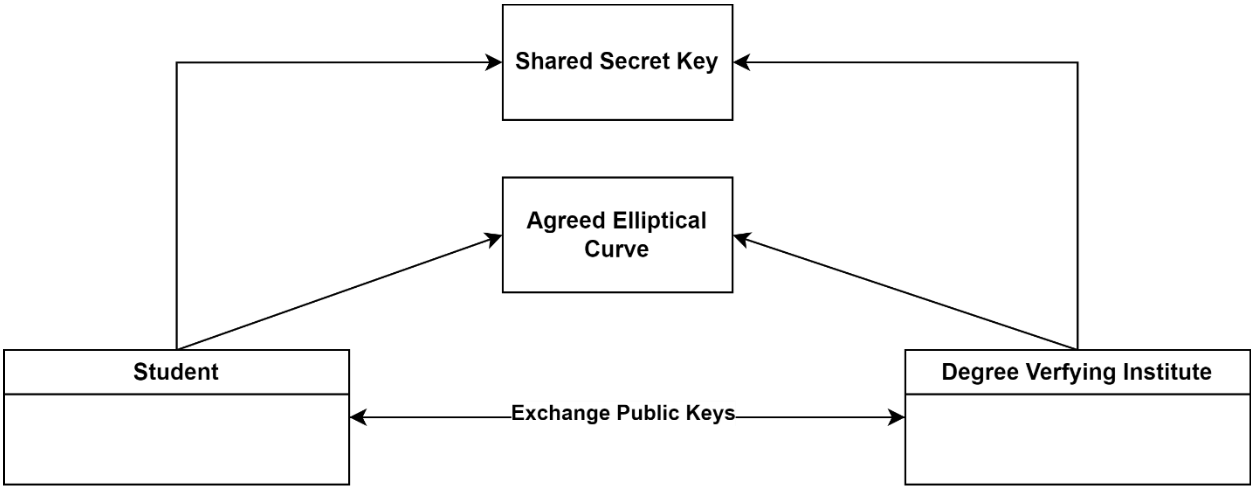
The student initiates the process by generating a random ECC key pair: {studentPrivKey, studentPubKey = studentPrivKey * G}.

The Verifying Institution follows by creating its ECC key pair: {institutionPrivKey, institutionPubKey = institutionPrivKey * G}.

The public keys of both the Student and the Verifying Institution denoted as studentPubKey and institutionPubKey, are made public

Figure 3

Shared Secret Key Generation using ECDH



As shown in Figure 3, the student calculates the shared secret key as:

$$sharedKey = institutionPubKey * studentPrivKey \tag{1}$$

Simultaneously, the Verifying Institution computes the shared secret key as:

$$sharedKey = studentPubKey * institutionPrivKey \tag{2}$$

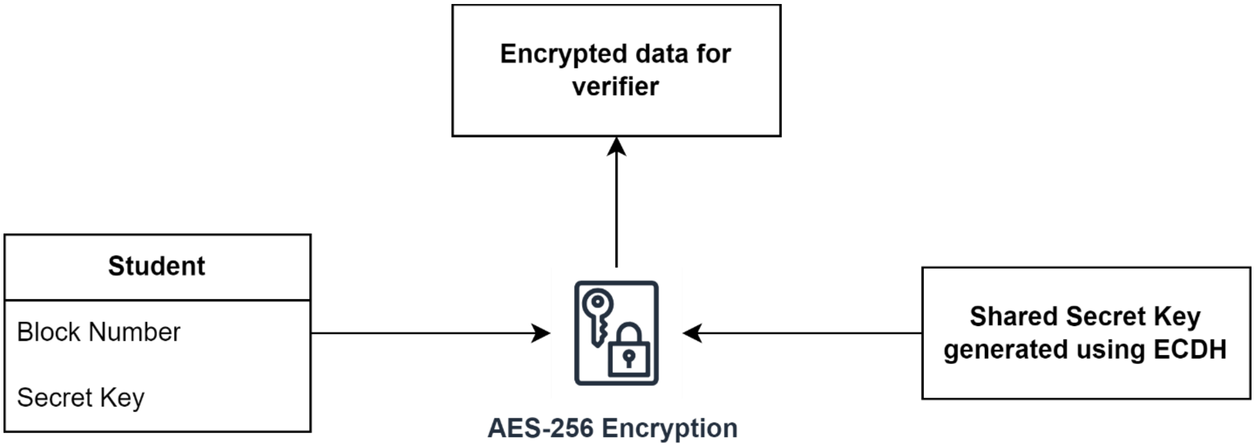
In this way, both the Student and the Verifying Institution successfully establish a secure, shared secret.

Step 5: Data Encryption and Transmission

As shown in Figure 4, the student leverages the shared secret key to encrypt the academic record's block number in the Ethereum network and the AES secret key, using AES-256 adding an additional layer of protection. The encrypted data, encompassing the block number and AES secret key, is then transmitted securely to the receiving institution via trusted communication channels. This encryption process adds an additional layer of protection, rendering the details indecipherable to any unintended recipients.

Figure 4

Encrypting Student's Ethereum block details

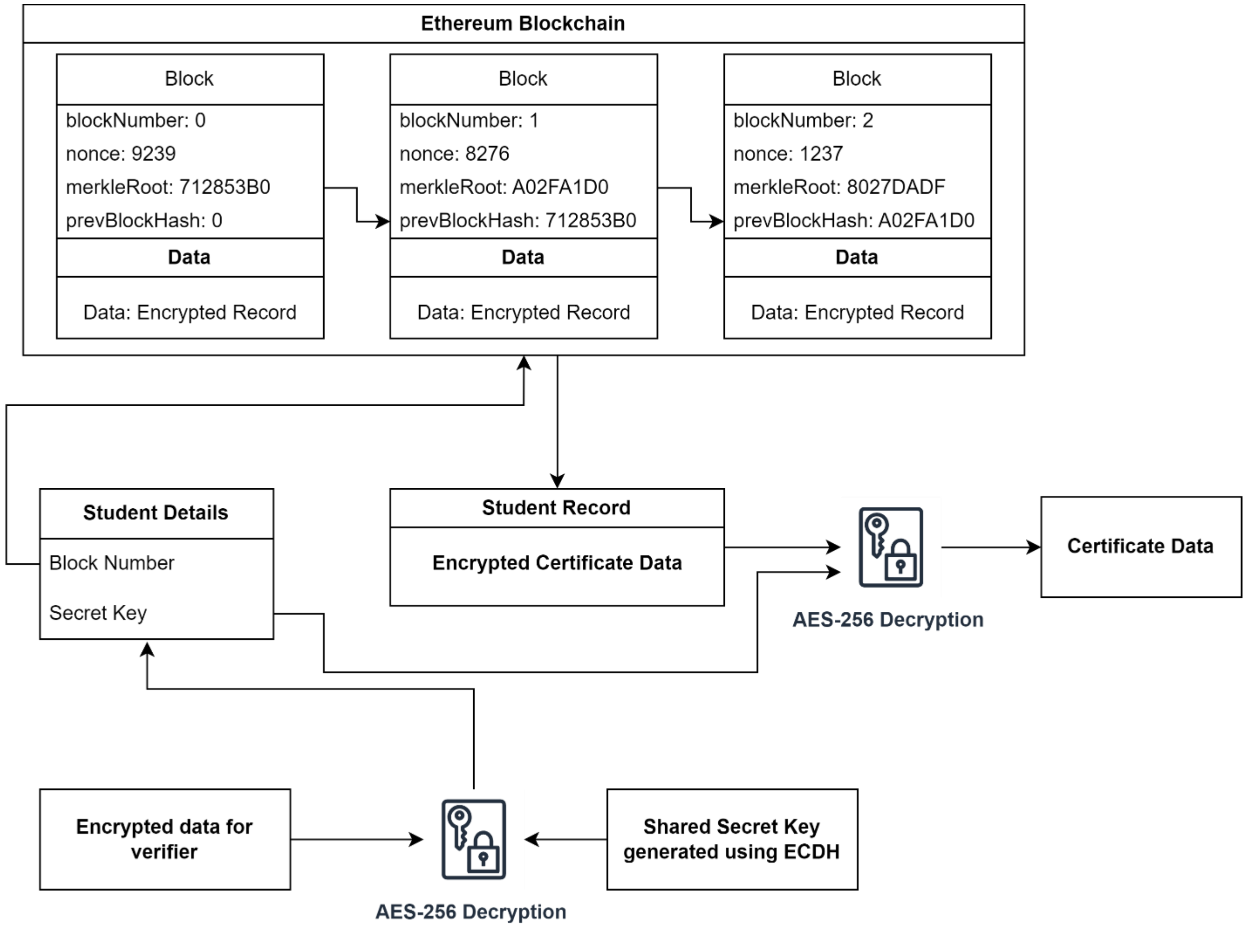

Encrypted data for verifier
Student
Block Number
Secret Key
AES-256 Encryption
Shared Secret Key generated using ECDH

Step 6: Data Decryption and Verification

In this step, the receiving institution holds the secret key to decrypt the encrypted details of the academic record, which was securely transmitted by the student. The receiving institution now has access to the student's academic record block number and the secret key to decrypt it.

Figure 5

Fetching Student Record from Blockchain and Decrypting the Record



To access the academic record on the Ethereum network, the institution requires access to two critical components: the Ethereum smart contract's ABI (Application Binary Interface) and its contract address. These components, as previously established, facilitate the interaction with the blockchain. The ABI acts as a crucial interpreter, bridging the gap between the Ethereum blockchain and the receiving institution's systems. It specifies how to communicate with the Ethereum smart contract, defining the functions and data structures that are available for interaction. The contract address, on the other hand, is the unique identifier pointing to the exact location of the academic record within the Ethereum network. Utilizing the ABI and contract address, the receiving institution accesses the Ethereum block. As shown in Figure 5, using the secret key that is shared by the student in the prior step, the institution decrypts the data academic record and verifies the academic record's authenticity. The blockchain's immutable ledger ensures the record's origin and integrity, providing an indomitable source of truth.

Throughout this exchange, data privacy and confidentiality are maintained. The use of ECDH and secure communication channels has safeguarded the academic record's confidentiality during transmission, further improving the system's commitment to privacy and security. This process guarantees that academic credentials are transferred, verified, and stored with the utmost integrity and confidentiality, thus preserving their credibility and trustworthiness.

Advantages

The implementation of advanced security measures forms the foundation of our academic records system. Utilizing AES-256 encryption and ECDH for key exchange ensures robust security, safeguarding academic records against unauthorized access and tampering. The integration with the Ethereum blockchain adds an extra layer of security, storing records immutably and creating a tamper-proof history of qualifications. The decentralized nature of Ethereum enhances transparency, as records are verified by a network of nodes, reducing the risk of manipulation. Additionally, the system remains quantum-resistant with the adoption of AES-256, ensuring long-term data security. The computational efficiency of ECDH allows for resource-efficient verification, making it suitable for various environments. The flexibility in security levels provided by ECDH, along with the efficient automation of verification processes through smart contracts, significantly reduces the time and effort required for academic credential verification, promoting data privacy during transmission.

Furthermore, the system prioritizes environmental sustainability by transitioning to Ethereum's Proof of Stake (PoS) consensus mechanism, minimizing energy consumption compared to the energy-intensive Proof of Work (PoW). This eco-friendly approach contributes to a reduced environmental impact. The global compatibility of the system, driven by blockchain technology and digital records, streamlines cross-border verification and facilitates the efficient transfer of credentials worldwide. The overall cost-efficiency of the system, achieved through automation and reduced resource requirements, benefits both educational institutions and students. Additionally, the design's future adaptability ensures long-term security, with the ability to adjust key spaces or encryption methods to address evolving threats and technologies.

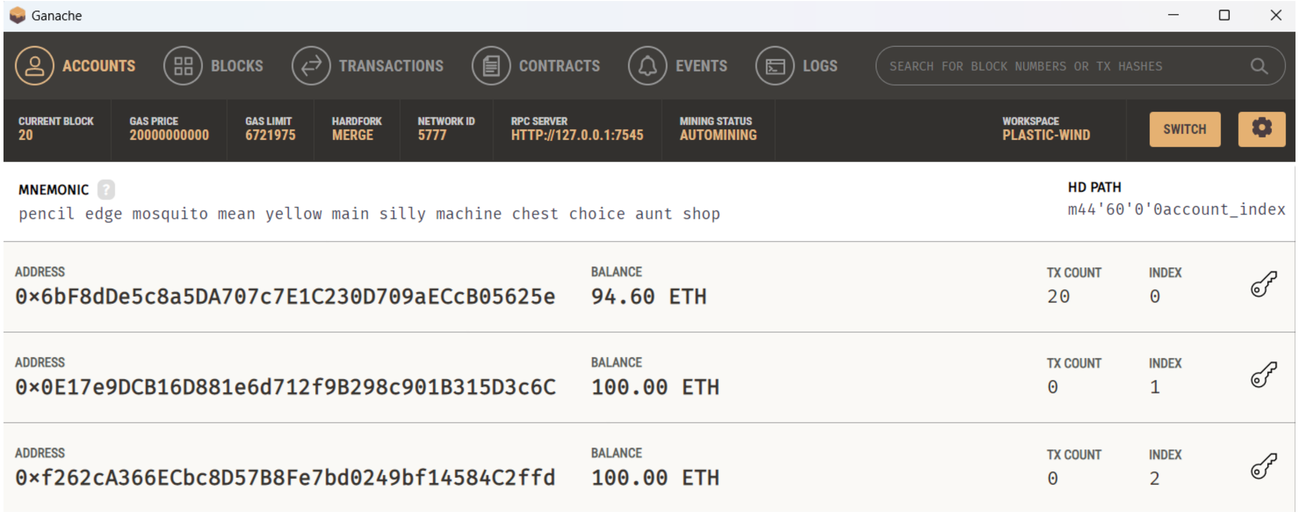
Challenges

One of the primary challenges in implementing a blockchain-based academic credential verification system involves securely exchanging elliptic curve parameters between users and institutions. Any compromise in this process could jeopardize the overall security of the system. Additionally, the fluctuating and sometimes high gas fees on the Ethereum network pose a financial challenge for educational institutions and students engaging with the system. Transaction costs associated with blockchain interactions, aside from gas fees, can vary based on the volume of academic records processed. Effectively managing these transaction expenses remains a continuous concern for the system's sustainability. Encouraging user adoption presents another hurdle, requiring institutions and students to transition from traditional credential verification methods to blockchain-based system.

Addressing regulatory compliance is a complex and time-consuming aspect of the implementation process. Ensuring the system complies with international and local regulations for academic credential verification, data protection, and privacy is crucial. Moreover, achieving data interoperability poses a challenge, given the diverse data structures and standards across different educational institutions. The challenge of ensuring data security while adhering to evolving encryption standards is essential for the long-term viability of the blockchain-based academic credential verification system.

Code

Tool 1: Ganache is a personal blockchain tool for rapid Ethereum distributed application development



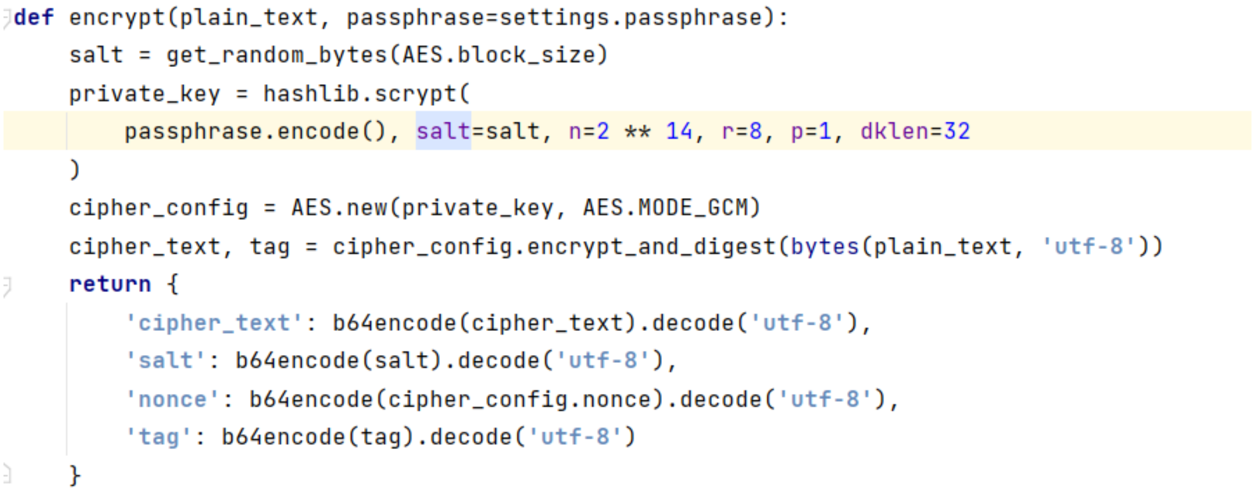
CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
20	20000000000	6721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	PLASTIC-WIND

MNEMONIC
pencil edge mosquito mean yellow main silly machine chest choice aunt shop

HD PATH
m44'60'0'0account_index

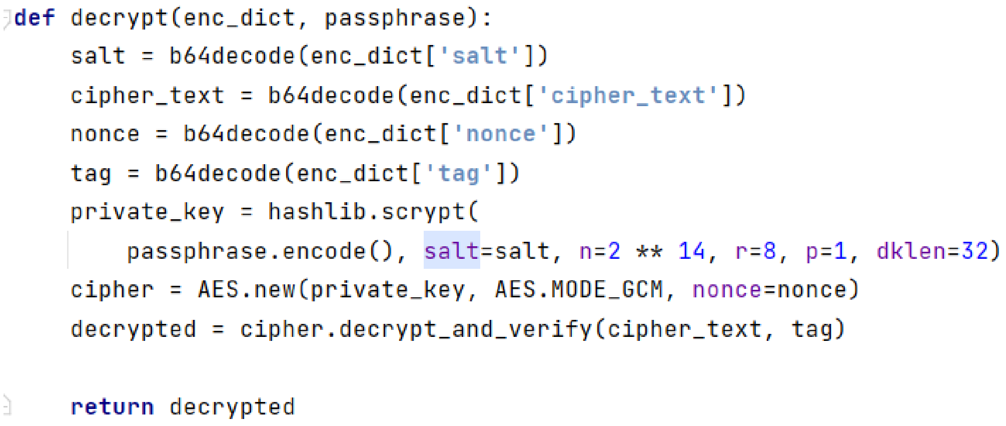
ADDRESS	BALANCE	TX COUNT	INDEX
0×6bF8dDe5c8a5DA707c7E1C230D709aECcB05625e	94.60 ETH	20	0
0×0E17e9DCB16D881e6d712f9B298c901B315D3c6C	100.00 ETH	0	1
0×f262cA366ECbc8D57B8Fe7bd0249bf14584C2ffd	100.00 ETH	0	2

Code 1: Encryption using AES 256 with GCM



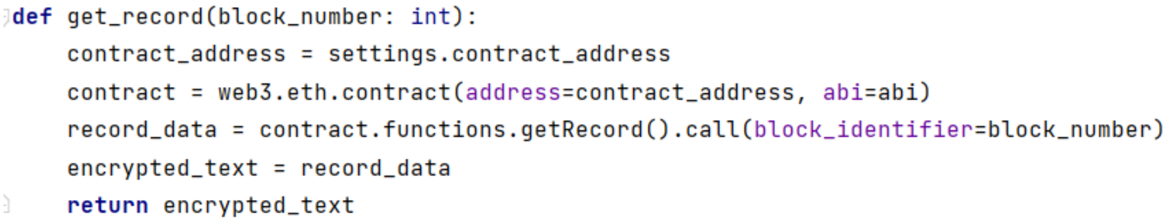
```python
def encrypt(plain_text, passphrase=settings.passphrase):
    salt = get_random_bytes(AES.block_size)
    private_key = hashlib.scrypt(
        passphrase.encode(), salt=salt, n=2 ** 14, r=8, p=1, dklen=32
    )
    cipher_config = AES.new(private_key, AES.MODE_GCM)
    cipher_text, tag = cipher_config.encrypt_and_digest(bytes(plain_text, 'utf-8'))
    return {
        'cipher_text': b64encode(cipher_text).decode('utf-8'),
        'salt': b64encode(salt).decode('utf-8'),
        'nonce': b64encode(cipher_config.nonce).decode('utf-8'),
        'tag': b64encode(tag).decode('utf-8')
    }
```

Code 2: Encryption using AES 256 with GCM



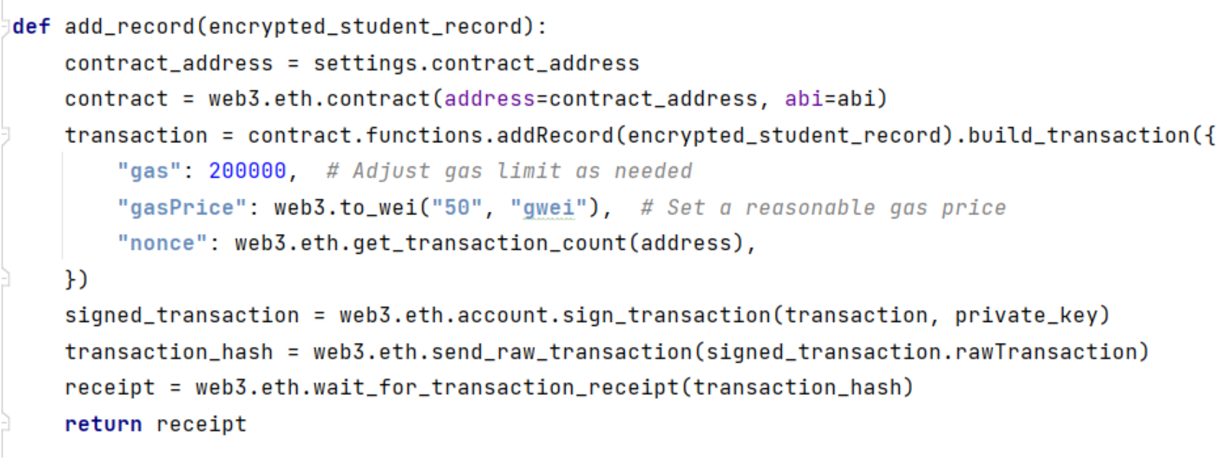
```python
def decrypt(enc_dict, passphrase):
    salt = b64decode(enc_dict['salt'])
    cipher_text = b64decode(enc_dict['cipher_text'])
    nonce = b64decode(enc_dict['nonce'])
    tag = b64decode(enc_dict['tag'])
    private_key = hashlib.scrypt(
        passphrase.encode(), salt=salt, n=2 ** 14, r=8, p=1, dklen=32)
    cipher = AES.new(private_key, AES.MODE_GCM, nonce=nonce)
    decrypted = cipher.decrypt_and_verify(cipher_text, tag)

    return decrypted
```

Code 3: Get record from Ethereum based on Block number



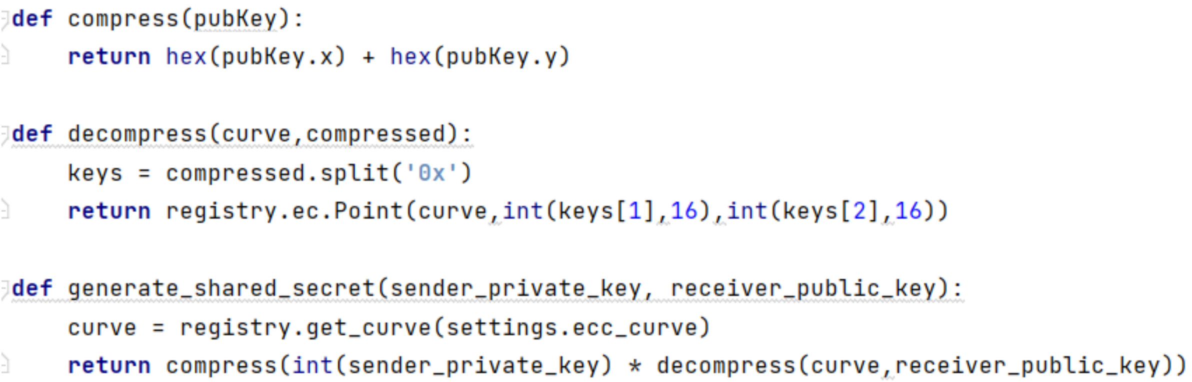
```python
def get_record(block_number: int):
    contract_address = settings.contract_address
    contract = web3.eth.contract(address=contract_address, abi=abi)
    record_data = contract.functions.getRecord().call(block_identifier=block_number)
    encrypted_text = record_data
    return encrypted_text
```

Code 4: Add record to Ethereum



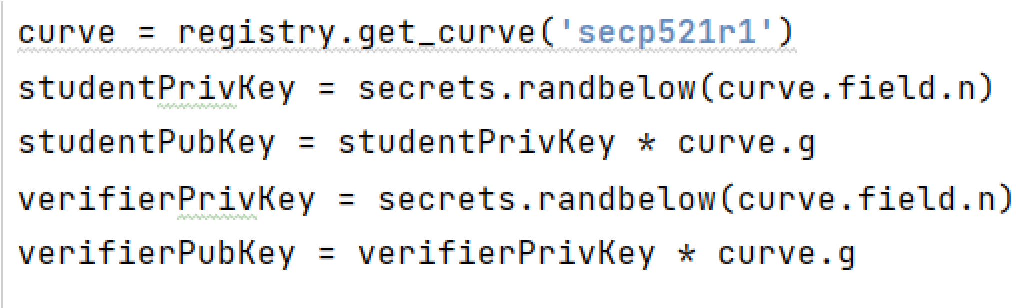
```
def add_record(encrypted_student_record):
    contract_address = settings.contract_address
    contract = web3.eth.contract(address=contract_address, abi=abi)
    transaction = contract.functions.addRecord(encrypted_student_record).build_transaction({
        "gas": 200000,  # Adjust gas limit as needed
        "gasPrice": web3.to_wei("50", "gwei"),  # Set a reasonable gas price
        "nonce": web3.eth.get_transaction_count(address),
    })
    signed_transaction = web3.eth.account.sign_transaction(transaction, private_key)
    transaction_hash = web3.eth.send_raw_transaction(signed_transaction.rawTransaction)
    receipt = web3.eth.wait_for_transaction_receipt(transaction_hash)
    return receipt
```

Code 5: Secret key generation using ECDH



```
def compress(pubKey):
    return hex(pubKey.x) + hex(pubKey.y)

def decompress(curve,compressed):
    keys = compressed.split('0x')
    return registry.ec.Point(curve,int(keys[1],16),int(keys[2],16))

def generate_shared_secret(sender_private_key, receiver_public_key):
    curve = registry.get_curve(settings.ecc_curve)
    return compress(int(sender_private_key) * decompress(curve,receiver_public_key))
```

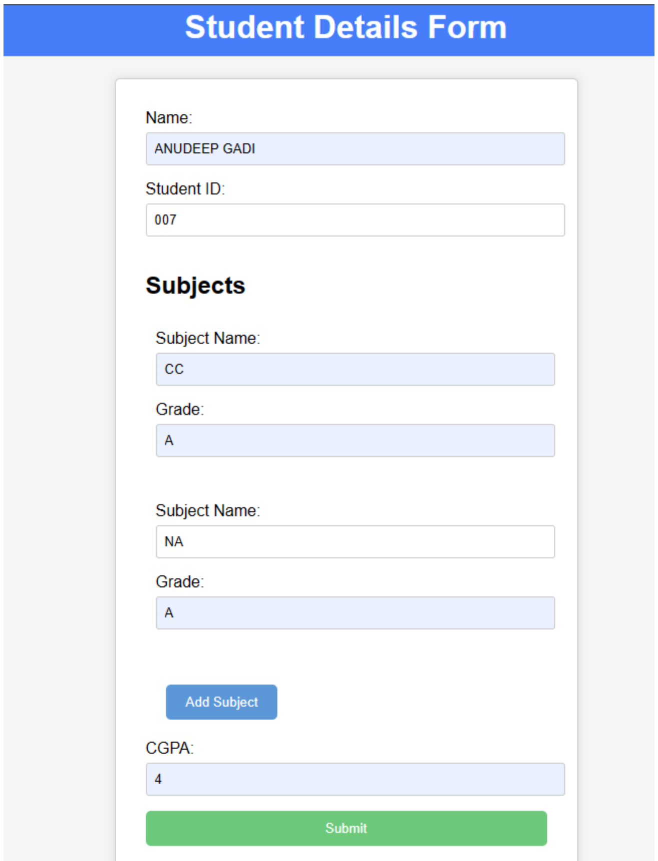
Code 6: Generating ECDH Private and Public keys



```
curve = registry.get_curve('secp521r1')
studentPrivKey = secrets.randbelow(curve.field.n)
studentPubKey = studentPrivKey * curve.g
verifierPrivKey = secrets.randbelow(curve.field.n)
verifierPubKey = verifierPrivKey * curve.g
```

Outputs

Output 1: Degree granting institution add details to Ethereum



Student Details Form

Name: ANUDEEP GADI

Student ID: 007

Subjects

Subject Name: CC

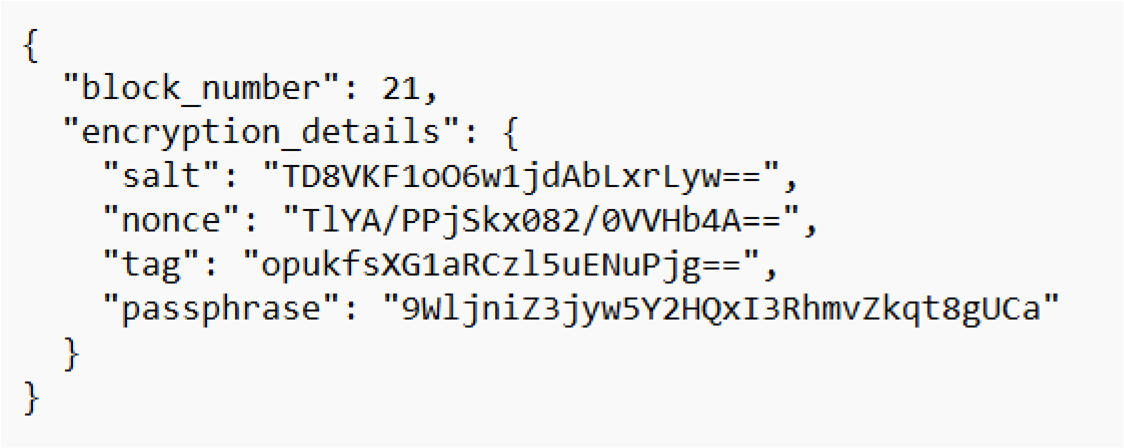
Grade: A

Subject Name: NA

Grade: A

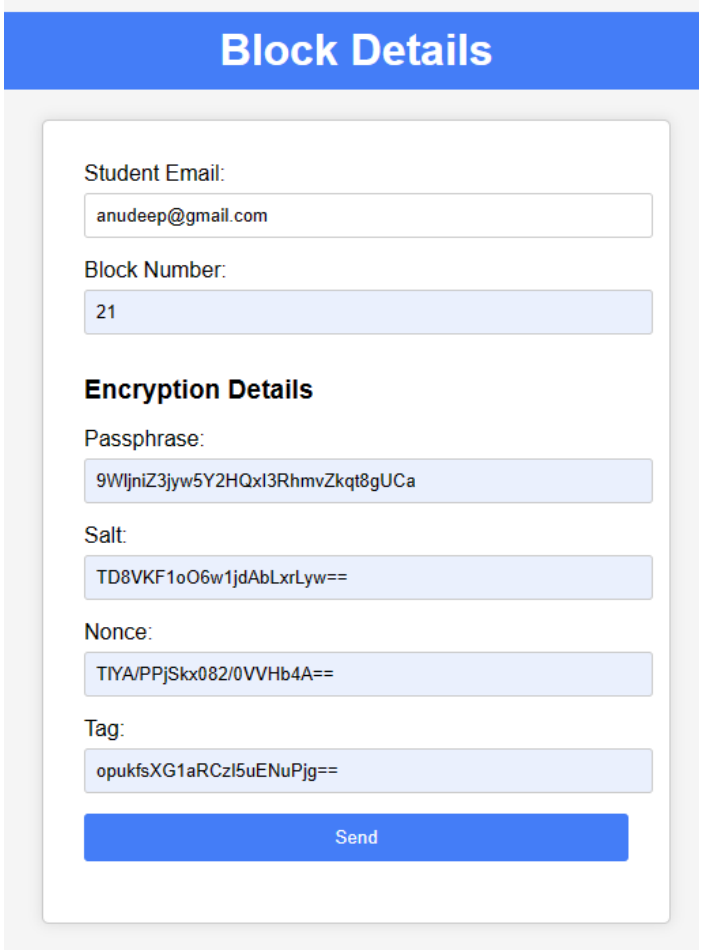
CGPA: 4

Output 2: Details that are to be sent securely to the student



```
{
  "block_number": 21,
  "encryption_details": {
    "salt": "TD8VKF1oO6w1jdAbLxrLyw==",
    "nonce": "TlYA/PPjSkx082/0VVHb4A==",
    "tag": "opukfsXG1aRCzl5uENuPjg==",
    "passphrase": "9WljniZ3jyw5Y2HQxI3RhmvZkqt8gUCa"
  }
}
```

Output 3: Student sending details to Verifying institution



Block Details

Student Email:
anudeep@gmail.com

Block Number:
21

Encryption Details

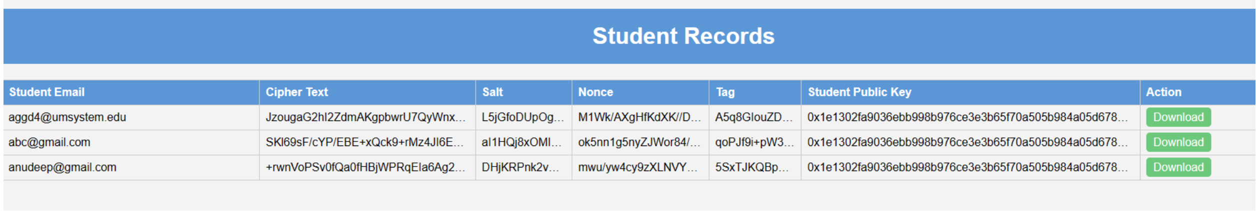
Passphrase:
9WljniZ3jyw5Y2HQxl3RhmvZkqt8gUCa

Salt:
TD8VKF1oO6w1jdAbLxrLyw==

Nonce:
TIYA/PPjSkx082/0VVHb4A==

Tag:
opukfsXG1aRCzl5uENuPjg==

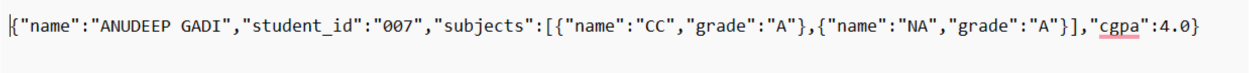
Output 4: Verifying institution verifying the records



Student Records

Student Email	Cipher Text	Salt	Nonce	Tag	Student Public Key	Action
aggd4@umsystem.edu	JzougaG2hl2ZdmAKgpbwrU7QyWnx...	L5jGfoDUpOg...	M1Wk/AXgHfKdXK//D...	A5q8GlouZD...	0x1e1302fa9036ebb998b976ce3e3b65f70a505b984a05d678...	Download
abc@gmail.com	SKl69sF/cYP/EBE+xQck9+rMz4JI6E...	al1HQj8xOMI...	ok5nn1g5nyZJWor84/...	qoPJf9i+pW3...	0x1e1302fa9036ebb998b976ce3e3b65f70a505b984a05d678...	Download
anudeep@gmail.com	+rwnVoPSv0fQa0fHBjWPRqEla6Ag2...	DHjKRPnk2v...	mwu/yw4cy9zXLNVY...	5SxTJKQBp...	0x1e1302fa9036ebb998b976ce3e3b65f70a505b984a05d678...	Download

Output 5: Verifying Records



```
{"name":"ANUDEEP GADI","student_id":"007","subjects":[{"name":"CC","grade":"A"},{"name":"NA","grade":"A"}],"cgpa":4.0}
```

Results

The proposed system is successfully implemented, employing Advanced Encryption Standard with Galois/Counter Mode (AES GCM) for robust data encryption and Elliptic Curve Diffie-Hellman (ECDH) for secure key exchange. Performance testing demonstrated satisfactory response times, striking a balance between heightened security measures and minimal impact on system efficiency. The average cost to send and verify transcripts using existing services is 217 USD as of 2023. In the proposed system, we incentivize the miners only when we are adding a record to Ethereum, we need not pay when we are reading from a block as we already know the block number. The system uses a gas of 2,00,000 with a price of 50 Gwei (a billionth part of Eth, 1 Eth equals 2,041.40 USD as of 2023) for each unit of gas. This value equals to 20.44 USD approx. as of 2023. The proposed system provides a cost-effective solution for academic record verification.

Furthermore, the proposed system does not just revolutionize cost efficiency but also significantly enhances the speed and accessibility of the verification process. Unlike traditional methods that often involve lengthy waiting periods, this system enables the verification of academic records with just a click of a button. This swift verification process eliminates the need for days or weeks of waiting, providing instant confirmation of academic credentials. By removing intermediaries like third-party verification services, this streamlined approach ensures a direct and seamless interaction between the institution and the blockchain, simplifying the entire verification journey. This not only accelerates the verification process but also makes it more user-friendly, marking a substantial advancement towards a faster, cost-effective, and decentralized method of academic record verification.

Conclusion

The integration of Ethereum blockchain, AES-256 GCM, and ECDH key exchange in the proposed academic credential verification revolutionizes the education sector. Leveraging Ethereum's decentralized architecture and the security of AES-256 GCM, the system ensures tamper-resistant and confidential record verification. The adoption of energy-efficient Proof of Stake further emphasizes the commitment to sustainability. Despite facing challenges in curve sharing, gas fees on the Ethereum network, and the looming threat of quantum computing, the system represents a significant leap towards a more secure, scalable, and efficient future for academic credential verification.

Moreover, this system not only excels in technological advancements but also champions environmental responsibility. By eliminating the reliance on paper-based documentation, the digitalized approach significantly reduces resource consumption and environmental impact associated with traditional verification processes. The transition to a paperless environment, coupled with the permanent and secure storage of records on the blockchain, not only ensures the integrity and longevity of academic credentials but also contributes to a greener, more sustainable future. In the face of challenges and complexities, the proposed system stands as a testament to the transformative power of technology in creating a secure, efficient, and eco-friendly landscape for academic credential verification.

References

P. Gugnani, W. W. Godfrey and D. Sadhya, "Ethereum Based Smart Contract for Event Management System," 2022 IEEE 6th Conference on Information and Communication Technology (CICT), Gwalior, India, 2022, pp. 1-5, doi: 10.1109/CICT56698.2022.9997939.

D. R. Rv, S. Mohamed Thawfiq., U. Arshad Uvais. and S. Revathy., "URL Protection and Bookmark Hiding using AES Algorithm," 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 766-771, doi: 10.1109/ICCES54183.2022.9836009.

T. Adhikari, M. Kule and A. K. Khan, "An ECDH and AES Based Encryption Approach for Prevention of MiTM in SDN Southbound Communication Interface," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-5, doi: 10.1109/ICCCNT54827.2022.9984509.

H. Gaikwad, N. D’Souza, R. Gupta and A. K. Tripathy, "A Blockchain-Based Verification System for Academic Certificates," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 2021, pp. 1-6, doi: 10.1109/ICSCAN53069.2021.9526377.

A. Singh, S. Chauhan and A. K. Goel, "Blockchain Based Verification of Educational and Professional Certificates," 2023 2nd International Conference on Computational Systems and Communication (ICCSC), Thiruvananthapuram, India, 2023, pp. 1-7, doi: 10.1109/ICCSC56913.2023.10143008.